COVER SHEET

DESIGN AND IMPLEMENTATION OF E1 STREAM ANALYZER FOR CCSS7 PROTOCOL

A.Y.K. Malik, M.M. Ahmed, A. Azhar and O.A. Khan

Faculty of Electronic Engineering, GIK Institute of Engineering Sciences and Technology, Topi, Pakistan.
E-mail ahmad@giki.edu.pk

Abstract— *E1 (PCM-30) Stream is a European Telecommunication standard for communication of data/voice channels in a digital network. This paper will deal with the design and implementation of a transparent stream analyzer, capable of handling (up to 2 Mbps) E1 stream and analyzing 30 voice /data channels using an out-of-band Common Channel Signaling System number 7 (CCSS7) protocol. The analyzer is capable of generating an E1 stream and thus can be used for generation, monitoring and blocking of channels. The Analyzer can perform analysis and maintenance of E1 (2.048Mbps) lines. As our analyzer performs the analysis functions in the software it is very flexible and can perform many additional features such as algorithms and protocols testing and analysis. The workable synchronous Finite State Machine (FSM) design of analyzer is implemented in Verilog and verified on Altera's FLEX10K70 FPGA along with its subsequent interfacing with PC through Enhanced Parallel Port.*

Keywords: *Pulse Code Modulation (PCM), Finite State Machine (FSM), Common Channel Signaling System number 7(CCSS7), Field Programmable Gate Array (FPGA).*

1. INTRODUCTION

The E1 stream is captured and synchronized by passing through the synchronous module and after that channel extraction module extracts the channel 16 (CCSS7) and pass it to Parallel Port module which transfers the data to PC where they are analyzed. All of these modules are FSM based and are controlled by a master controller which also helps in establishing communication with the PC. This paper will deal with the design and implementation of these modules in Verilog.

a) E1 (PCM-30) data/voice Stream

Links between Local Exchanges (LEs) and the backbone network and links between backbone switches (also known as "trunks) carry many PCM E1 streams using Time Division Multiplexing (TDM) [1]. TDM combines 64 Kbps voice streams to higher order bandwidth streams, with the aggregate link bandwidth being a function of the number of voice streams the link carries. A synchronous Time Division Multiplexer has n input links and one output link, which are at least n times faster. Each set of n samples with overhead bits for synchronization constitutes a frame. A receiver can extract (de-multiplex) each constituent stream by synchronizing to frame boundaries as in [2].

The Nyquist theorem state that for converting an analog to digital signal, sampling is required at a minimum rate twice that of the highest frequency component. Hence, to convert a 4 KHz voice signal to a bit stream, it must be sampled 8000 times per second. Sampling voice 8000 times / sec with 8 bit samples, leads to a standard voice bandwidth of 64 Kbps (one sample every 1/8000 = 125 μ sec). The resulting digital signal is known as Pulse Code Modulated (PCM) signal and it is the minimum unit of transmission capacity [3]. In the ITU-T system, a frame consists of 32\ 8-bit slots; with time slot 0 being used for synchronization and time slot 16 is used for signaling CCSS7 (8-bit/125 μ sec = 64 Kbps). This yields aggregate rate of 32*8 bits / 125 μ sec = 2.048 Mbps and the link is known as E1 as in [4], [5].

b) Common Channel Signaling System 7 (CCSS7)

Administered by ITU-T CSS7 common signaling system No. 7 is the out of band signaling channel. Out-of-band or common channel signaling is like a packet network riding herd on a circuit network [6]. With CSS7, the common channel protocol used in the telephone network, the originating exchange can communicate with the terminating exchange, computer to computer, before calls are set up [7], [5]. This allows efficiency and flexibility in the use of network resources. A new set of features can now be offered across the network, such as caller ID and ISDN. The common channel can identify the 64kb channel as voice or data, allowing end-to-end ISDN and other digital services. Signaling System 7 (SS7) is architecture for performing out-of-band signaling in support of the call-establishment, billing, routing, and information-exchange functions of the public switched telephone network (PSTN). It identifies functions to be performed by a signaling-system network and a protocol to enable their performance [8].

As we have moved towards convergence between the public circuit-switched telephone network and the packet-switched IP world, SS7 has become the subject of significant attention as developers seek to integrate the two worlds and leverage the best of both. CSS7 is symbiotic to PSTN as well.

CSS7 engages out-of-band signaling mechanism, now what does it mean, we have two categories of data on E1 links connecting various tandem exchanges and local exchanges, one is actual voice data or text data other is call management information data or the information required to set up a call between two subscribers, so CSS7 carries call management data of all the voice channels in the E1 link, making the circuit utilization more efficient and fast. CSS7 is based upon message switching; where IAM's (Initial Access Messages) are created and switched between various points known as STP (Signal Transfer Point), SCP (Service Control Point) and SSP (Service Switch Point). So the management data is in the form of the messages (packets) which are switched and routed b/w these various points instead of engaging public circuits, which can now be utilized for actual call data as in [8]. The STP (Signal Transfer Point) is the digital sister of a switch.

c) Architecture of CCSS7

In common with many signaling protocols, SS7 is made up of a layered architecture. Each layer has a specific role and responsibility. The lowest 3 layers together form the Message Transfer Part or MTP [9]. This is responsible for the secure and reliable routing of messages, the content of which is provided by other, higher layers. MTP uses signaling links to route messages to the required destinations. Higher layers have different functions and are implemented as required by the network. Call control (i.e. the establishment and disconnection of calls) is handled by one of a series of Layer 4 Call Control Protocols, such as ISUP or TUP as described in [10]. Other functions are built on top of another layer called SCCP [11]. Fig.(1) depicts the OSI [12] equivalent model of the SS7 architecture.

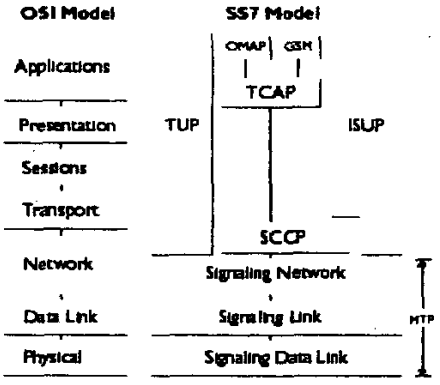

Figure 1 OSI and SS7 protocol.

d) Hardware aspects of analyzer

Analyzer design intends to capture the E1 stream and extract the information of our interest. The hardware used to capture the stream at this fast rate s Altera's FLEX 10K70 FPGA operating at a clock of 25.175 MHz [13]. The whole system has been synchronously designed. The main objective of this system is to either generate an E1 stream or capture a coming stream and pass it through the system and capture or transmit any channel on the stream. In Reference [14] the emphasis is on rapid prototyping where this design can also be used for such purposes.

Modules are the basic block of any hardware related project. They have special functions and characteristics and when combined together they form a system. The best way to design any system is to divide it in to different parts according to their functionality and then build them separately and test them independently. After the testing and verification of independent modules they are integrated to form the system. Then the whole system is tested according to its requirements. This technique of system development is known as bottom-up approach. The designing of this whole system has been modularized with each module doing its intended purpose. The whole system has been tested through simulations and verified by running it on the hardware.

e) ALTERA FLEX EPF10K70

The EPF10K70 device is based on SRAM technology. It is available in a 240-pin RQFP package and has 3744 logic elements (LEs) and nine embedded array blocks (EABs). Each LE consists of a four-input LUT, a programmable flip-flop, and dedicated signal paths for carry-and-cascade functions. Each EAB provides 2048 bits of memory which can be used to create RAM, ROM, or first-in first-out (FIFO) functions. EABs can also implement logic functions, such as multipliers, microcontrollers, state machines, and digital signal processing (DSP) functions. With 70,000 typical gates, the EPF10K70 device is ideal for intermediate to advance digital design systems [13].

The EPF10K70 device can be configured in-system with ByteBlasterMV download cable. Additionally, Maxplus II software provided by Altera is used to synthesize the design in Verilog and is then programmed in the FLEX device. The different pins are also assigned to the device using the same software. The board used for this project is Altera University Design laboratory Package which contains the above mentioned device and its associated programmer.

2. HARDWARE DESIGN OF ANALYZER

Proposed analyzer design is meant to capture, generate and analyze any voice/data channel and has got the capability to communicate with computer. The design is synchronous and all the modules work on the system clock. The basic line decoder such as Zarlink's MT9075b is assumed to be present so that a TTL level 2Mbps E1 Stream and a 2 MHz Clock are input to our FPGA and E1 stream is the output of our hardware. The main system hardware design aspects are discussed.

a) Main core of the system

In the implementation of the analyzer the whole system has been divided in to different modules according to its functionality. The main objective of the system is to be capable of generating E1 stream and also of capturing the stream in addition to placement and extraction of any channel. For this purpose there are several modules involved with a well defined structure and functionality. The main core is shown in Fig. (2)

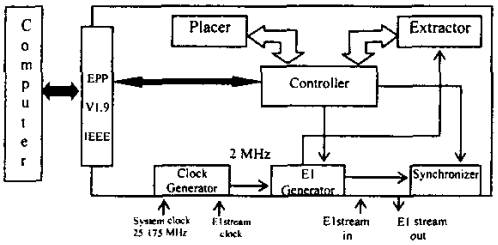

Figure 2 Main core of the analyzer

b) Synchronization Module

E1 stream consists of 32 channels with the time slot 0 (channel 1) being the synchronization channel. The purpose of this channel is to synchronize the whole stream so that each channel can be well defined. In E1 stream a known pattern 9D (hex) is being transmitted on the synchronization channel after every alternate frame of 32 channels **[4]**. In order to synchronize the stream we have to search for this 9D (hex) pattern in every alternate frame of 32 channels. Once the pattern is matched it keep on checking for the synchronization and whenever it is lost it will tell the whole system that synchronization is lost and will start searching for synchronization again.

The whole module of synchronization is implemented in a FSM shown in Fig.(3). The FSM has got 3 states where:

- State-1 (Not Sync) is the state where no synchronization is achieved and the search for Frame alignment pattern is going on. When ever the pattern is found it will initiate a counter and move to state2.
- State-2 (Pattern Found) is the state where pattern is detected and if another pattern after the 9-bit counter is fully counted is received then it will move to state3 which is synchronized state. If after the counter has fully counted and the pattern is not received than it will go back to state1.
- State-3 (Stream Synchronized) is the state where the stream is synchronized and it will keep on checking the incoming stream for synchronization and whenever the synchronization is lost it will go back to state1.

In Fig.(4) the flow diagram shows the pragmatic working of the module. The E1 stream enters the shift register where the extracted clock of the demodulator (not shown) provides the clock to the shift register. The out put of the shift register is the output E1 stream and helps the analyzer to be transparent, i.e. the analyzer is invisible to the system.

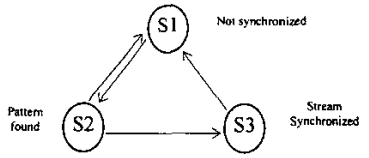

Figure 3 Shows state diagram for stream synchronization

The shift register compares the pattern to our desired 9D(hex) pattern and when it recognizes the information it starts the counter and as the next pattern appears after every alternate frame when counter counts up to 512 and the pattern is also there than it will tell the system that the stream is synchronized. If synchronization is lost it will again go to state 1 and try to again synchronize the stream. The inherent ability of the E1 trunk to automatically synchronize itself to incoming data is a key factor for its vide application in telecommunication sector.

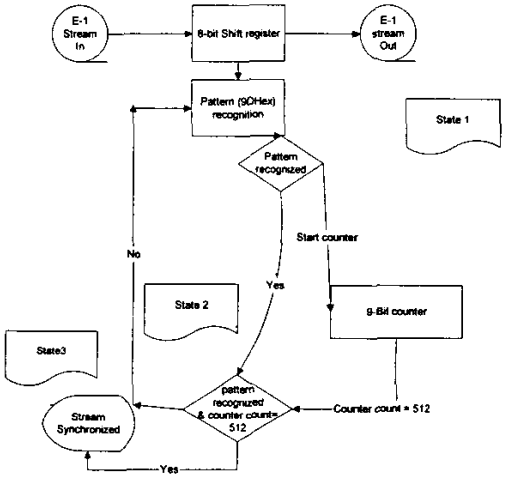

Figure 4 Flow diagram of synchronization module

c) Extractor and Placer Module

Once the stream is synchronized, each channel location from the counter value is known and one can extractor or place the information accordingly. The data is extracted or placed once the particular timeslot signal is received from a sub-module known as 'Timing module'.

Working of this module is also implemented on state machine where input to this state machine is the counter values which when matched with the prescribed timeslot value it will store the value at that particular instant in timeslot data register in case of extraction and if write enable is true that it will place the information on the

channel. The functionality of this module is shown in Fig. (5).

d) Timing Module

This module is a sub-module of Extractor and Placer module and it gives a pulse of a particular channel when ever that channel data is received. This module take counter value as input and when the stream is synchronized it starts giving the pulses of different timeslot on there prescribed output. This module take counter value as input and when the stream is synchronized it starts giving the pulses of different timeslot on there prescribed output.

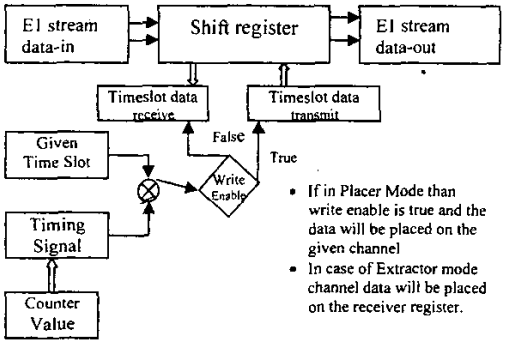

Figure 5 Shows placer and extractor module of an E1 stream analyzer.

These timing signals are used throughout the system. For instance, in EPP module they are used to place the required channel information for delivery to computer.

e) Clock Generator Module

In this module a 2MHz clock has been generated from a 25.175 MHz clock. The technique employed is that a counter runs at a frequency of 25.175 MHz and counts up to the value of 6 and toggles a single bit register. This register value will give a clock of 50% duty cycle at 2 MHz required frequency.

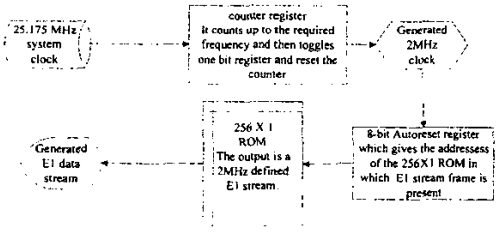

Figure 6 Shows clock and E1 stream generator module implemented in FPGA

f) E1 stream Generator Module

In this module there is a ROM of capacity 256x1 placed in an FPGA. The input address of this ROM is an 8-bit auto-reset counter which is incremented by a clock of 2 MHz. The data in the ROM is predefined. As 256 bits are equal to one frame of E1 stream, so every eight bit represents the information in a channel. Once this module start working, than the output of the ROM is taken as an E1 stream input as shown in Fig.(6). Once this is done the placer module can be used to place any information on any channel.

g) EPP Module

This module is also implemented upon state machine principle and is implemented around IEEE 1284 recommendations (V1.9) whose details are given in **[15]**. The module can read and write address and data from host EPP port controller such as PC's Parallel port. This module remains in the idle state and when it is given data/address to send to PC it will engage two out of three spare bits and in this way PC will know that the device wants to send data and will perform the required handshaking. During this handshaking the module will change the state to either data send state or address send state and will go back to idle state when the handshaking is over. Similar is the case with receiving data or address. The controlling device will put the data in the module by going into the read address or data state and the module will in turn engage the third spare bit to show that it is processing the information and no further data should be sent. The module operates at a frequency of 25.175MHz which is enough to provide the required handshaking timing requirements. There are certain bits which are used to indicate to the system that the data has been received or sent so that the system can make use of the information. A brief state diagram of EPP module is shown in Fig.(7).

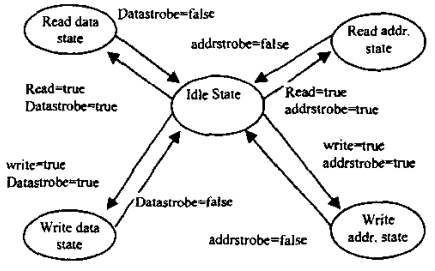

Figure 7 State diagram of EPP V1.9 Module

h) Main controller module

The main head of all modules is controller. This module interacts with different modules and makes them into the formation of system. This stands at the top in the system hierarchy. This module is used to place the system in generation or analyzer mode along with other options.

This module also talks with the PC which is the main controlling unit in Analyzer. It has got the capacity to send either a single channel or all the channels to the PC and it can also place the information on single or multiple channels. The Main controller modules can be set in different modes where the analyzer can act as stream generator or stream capturer. In the analyzer hardware design the controller can pass any channel to the computer and the usage of EPP port makes it possible to transmit the whole E1 stream to the computer. In such a mode the analyzer can be used as an End Point Local Exchange. Controller module is also state machine based where each option represents a state. The state can switch to any other state depending upon the conditions. If the stream is not synchronized than this information is sent to the PC and no analysis is made.

3. EXPERIMENTAL RESULTS

For testing purposes two Altera's University Package boards, one in the form of E1 Stream Generator and the other in the form of capturer/analyzer were employed. The CCSS7 data packet was generated in the PC and placed on the generator's channel 16 reserved for transmission of signaling information **[16]**, **[17]**. The analyzer extracts the CCSS7 packet, from PCM 30 E1 stream and passes it on to the computer through EPP port where the Packet is analyzed.

The whole design is synchronous and FSM based which makes debugging in the hardware convenient. One of the simulation timing diagrams of our core is shown in Fig.(8) **[18]**.

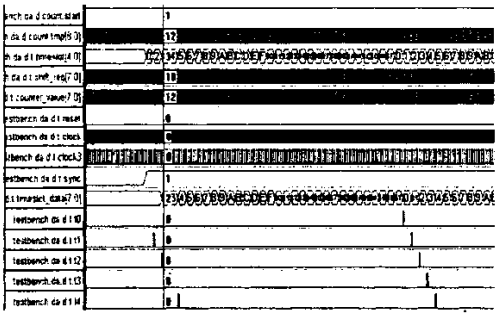

Figure 8 Timing diagram of timing & Synchronizer module.

In Fig.(8) it is shown that when the stream is synchronized i.e., the sync bit becomes high, the timeslot data is available at respective timing signals. In this timing diagram the data of each channel can be extracted at its respective timing signal.

4. APPLICATIONS

The proposed designed architecture relies heavily on the software where all the analysis is accomplished. Using a portable analyzing device (laptop) the analyzer can be made portable. The software dependence of our analyzer makes it flexible and general purpose analyzer. From computer the analyzer can be set to operate in different available mode as explained below:

- By default the analyzer transfers the Channel 16 (SS7) to computer which analysis the signaling information.
- In stream generator mode the analyzer can generate the E1 stream and signaling as well as voice/data is possible to be transmitted on the E1 link. This can also be used to debug the E1 links for errors.
- If placed between two E1 links the analyzer can act as a transparent device and thus can be used for monitoring of signaling as well as voice/data information between two exchanges.
- The ability of the analyzer to write on any given channel makes it possible to block or transmit information on any channel.
- Different encryption decryption algorithms can be tested and verified on the analyzer.
- As the analysis is software intensive the analyzer can be used to test the new protocols and standards by just changing the software.

5. CONCLUSION

The design of analyzer is verified by Altera's MAXPLUS II software and then loaded on the FLEX10K70 FPGA. Two prototype analyzers were built one for generation of stream and the other for analyzing of the stream. The signaling information (SS7) packet was placed on channel 16 of the generator and was successfully received and analyzed on the other end. Standard reusability of code practices are adopted in the design and implementation of Verilog modules and thus the system core can be employed in any other device.

ACKNOWLEDGMENT

The technical support about the project by Mr. Sayed Fayyaz Gilani and Mr. Yousuf Ali is highly acknowledged.

REFERENCES

[1] ITU-T Recommendation G.701, '*Vocabulary of digital transmission and multiplexing, and pulse code modulation (PCM) terms*', approved in 1993.

[2] ITU-T Recommendation G.702, '*Digital hierarchy bit rates*', 1988.

[3] ITU-T Recommendation G.711, '*Pulse code modulation (PCM) of voice frequencies*', 1988.

[4] ITU-T Recommendation G.732, '*Characteristics of primary PCM multiplex equipment operating at 2048 kbit/s*', 1988.

[5] Grahame Smillie, '*Analogue and Digital Communication Techniques*', pages 100-124, Published by Arnold 1999.

[6] ITU-T Recommendation Q.700, '*Introduction to CCITT Signalling System No. 7*', 1993.

[7] ITU-T Recommendation Q.705, '*Signalling network structure*', 1993.

[8] ITU-T Recommendation Q.110, '*General aspects of the utilization of standardized CCITT signalling systems on PCM links*', 1988.

[9] ITU-T Recommendation Q.701, '*Functional description of the message transfer part (MTP) of Signalling System No. 7*', 1993.

[10] ITU-T Recommendation Q.721, '*Functional description of the Signalling System No. 7 Telephone User Part (TUP)*', 1988.

[11] ITU-T Recommendation Q.716, '*Signalling System No. 7 - Signalling connection control part (SCCP) performance*', 1993.

[12] Technology series by Sunrise Telecom, '*Introduction to Signaling system No. 7*', CA, USA

[13] Altera Corporation's FLEX 10K70 FPGA, embedded in University Design Laboratory Package UP2 board.

[14] A. Vasilliou, K. Gounaris, K. Adaos, G. Alexiou, D. Nikolos, D. Mitsainas, paper on '*Development of a Reusable E1 Transceiver Suitable for Rapid Prototyping*', presented in Tenth IEEE International Workshop on Rapid System Prototyping, 1999, Clearwater, Florida.

[15] IEEE Std. 1284-1994, "*Standard Signaling Method for a Bi-Directional Parallel Peripheral Interface for Personal Computers*", 1994.

[16] ITU-T Recommendation Q.780, '*Signalling System No. 7 test specification - General description*', 1995.

[17] ITU-T Recommendation Q.713, '*Signalling connection control part formats and codes*', 2001.

[18] A.Y.K. Malik, A. Azhar, O.A. Khan, "*E1 Stream Analyzer, Capturing of CSS7*", Senior Design project report 2003, FEE, GIK Institute, Topi